UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/840,069	05/04/2004	Michael L. Boucher	30014200-1110	6646

58328 7590 03/19/2008

SUN MICROSYSTEMS
C/O SONNENSCHEIN NATH & ROSENTHAL LLP
P.O. BOX 061080
WACKER DRIVE STATION, SEARS TOWER
CHICAGO, IL 60606-1080

EXAMINER	
DENG, ANNA CHEN	
ART UNIT	**PAPER NUMBER**
2191	

MAIL DATE	DELIVERY MODE
03/19/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/840,069	BOUCHER, MICHAEL L.	
	Examiner	Art Unit	
	ANNA DENG	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 December 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to amendment filed on 12/17/2007.

2. The objection to the specification is withdrawn in review of applicant's amendment.

3. The rejection under 35 U.S.C. 101 to claims 7-12 and 16 is withdrawn in review of applicant's amendment.

4. Claims 1-16 are pending.

5. Claims 1-16 stand finally rejected.

Response to Amendment

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

The amended independent claims 1, 7, 13 and 16, recite "program code of the program". The specification recites "In developing software, typically a software developer writes code, referred to as "source code," in a programming language, and the source code is compiled by a compiler into "object code" that can be run by a machine. This code is also referred to as executable code (Specification, p1, lines 18-21, emphases added)". It is unclear that "program code of the program" is the "source code" written by the developers, or the "object code" compiled by the compiler. Therefore, the claim language of "program code of the program" is indefinite. Claims 2-6, 8-12, and 14-15 are rejected because their dependence on the base claims.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 1-16 are rejected under 35 U.S.C. 102(b) as being anticipated by USPN 6,381,735 B1 to Hunt.

Per claims 1,7, 13, and 16, (currently amended):

Hunt discloses:

- **A method (system / computer-readable medium) in a data processing system for optimizing a program** (Hunt, Abstract, "Dynamic classification of sections of software using a profile-based optimization system optimizes management of the sections of software"**, the method comprising the steps of:**

- **automatically analyzing a statistical profile of a program execution** (Hunt, col. 4, lines 18-21, "The present invention pertains to classification of a section of software by matching the usage background of the section to a usage profile (statistical profile) determined by previous profiling software"; col. 27, lines 35-55, "In general, through scenario-based profiling or static analysis, the illustrated ADPS creates a profile for each application unit instantiated. The profile classifies the application unit in order to characterize the application unit's communication with other units

during profiling and any constraints on its location. Information from the profiling scenarios or static analysis is generalized to predict application behavior for later executions. A mapping of generalized application unit profiles to specific machines in the network is generated. Application units instantiated during application execution are then matched to similar application unit profiles, and located on the appropriate machine in the network. The actual distribution is an approximate solution to the distributed partitioning problem: the optimal solution for a particular application execution can only be determined after execution has completed. The underlying assumption of automatic distributed partitioning is that past profiles are statistically accurate in describing future application executions. If, in fact, past profiles accurately predict future application executions, then future executions can be partitioned using the distribution derived from the profiles"); **and**

- **automatically optimizing program code of the program based on at least one of the analysis, information about at least one prior compilation of the program, and information about at least one prior optimization of the program** (Hunt, col. 27, lines 35-55, "In general, through scenario-based profiling or static analysis, the illustrated ADPS creates a profile for each application unit instantiated ... Information from the profiling scenarios or static analysis is generalized to predict application behavior for later executions"; col. 28, lines 24-37, "A component can be dynamically classified at the time of instantiation using available contextual information such as the execution call stack, the arguments to the instantiation

function, and the identities of previously instantiated components"; col. 30, lines 42-67 through col. 31, lines 1-25, "Although the foregoing techniques for dynamic classification have been described with reference to classification of application units in an automatic distributed portioning system, these techniques are applicable outside of the field of automatic distributed portioning... For example, in the field of memory heap management, ... In the field of freeing unused memory...To balance the load among system resources in a computing system, a lifetime prediction for a program or data segment can be used to optimally allocate system resources...In general, the foregoing techniques for dynamic classification are applicable to profile-based optimization systems … Generally, there are three classes of solutions to accomplish this task according to the present invention: modify the application's source code, modify the application's binaries prior to execution, or manipulate the application's execution through run-time intervention. … Referring to code fragment 280 using call replacement in application source code, calls to the COM instantiation functions can be replaced with calls to the instrumentation by modifying application source…Using call replacement in application binary code (281), calls to the COM instantiation functions can be replaced with calls to the instrumentation by modifying application binaries…" emphases added).

Per claims 2, 8 (currently amended), and 14:

Hunt discloses:

- **the program is automatically optimized during a compilation** (Hunt, col. 16, lines 34-51, "An application description set 220 describing the behavior of the application is prepared at step 210 for the application 200. The application description set 220 can be supplied by an external source that analyzes the application 200 in advance, or can be generated by the illustrated ADPS itself. The application description set 220 can include static and/or dynamic metadata describing the application. For example, in the COIGN system, the application description set 220 can include static metadata derived from metadata provided by a Microsoft IDL compiler (MIDL). Alternatively, the application description set 220 can include static metadata generated by the illustrated ADPS through static analysis techniques. Dynamic analysis techniques can be used by the illustrated ADPS to include dynamic metadata (such as dynamic descriptions of units, descriptions of actual inter-unit communication between the units of the application 200, and descriptions of how much time was spent in each unit in computation) in the application description set 220").

Per claims 3, 9 (currently amended), and 15:

Hunt discloses:

- **the program is automatically optimized during a run-time compilation** (Hunt, col. 17, lines 16-26, "A distribution scheme 50 is the result of applying the environment description set 230 to the application description set 220. The distribution scheme 250 includes a mapping of application units to locations in a distributed computing environment. The units can be classified using static metadata

of the units. Alternatively, where run-time profiling was used to dynamically describe the units, the units can be classified according to dynamic behavior. At run-time, units of the application 200 are mapped using the distribution scheme 250 for location on an appropriate computer in the distributed computing environment").

Per claims 4, and 10 (currently amended):

Hunt discloses:

- **interrupting the program** (Hunt, col. 31, lines 45-57, "Borrowing from debugger techniques, breakpoint trapping of the COM DLL (284), instead of replacing the DLL, inserts an interception mechanism into the image of the COM DLL after it has been loaded into the application address space. At run time, the instrumentation system inserts a breakpoint trap at the start of each instantiation function. When execution reaches the function entry point, a debugging exception is thrown by the trap and caught by the instrumentation system. The major drawback to breakpoint trapping is that debugging exceptions suspend all application threads. In addition, the debug exception is caught in a second operating-system process. Interception via break-point trapping has a high performance cost");

- **recording an execution state of the program** (Hunt, col. 1, lines 24-32, "Through scenario-based profiling, an ADPS can create a profile for each application unit instantiated during profiling runs of the application. The profile identifies and quantifies communication between the application unit and other units. The

collection of profiles for all units in the application, together with the records of communications between units, can be included within the application description set 220 and used to decide where units should be placed in the network").

- **storing the recorded execution state to create the statistical profile** (Hunt, col. 19, lines 12-23, "At step 204, the application 200 is executed in a scenario meant to model the expected use of the application 200. During execution, the application behaves normally while the numbers, sizes, and endpoints of all inter-unit messages are measured. At step 206, the user decides if profiling is finished. The application can be run through an arbitrary number of profiling scenarios. After profiling of the application is completed, the results from the scenario-based profiling are written (step 208) to the application description set 220. The application description set 220 can include structural description of the application as well as description of communication between units of the application." col. 19, lines 51-55, "The environment description set 230 can be generated immediately before the application is to be distributed in a distributed computing environment, in this way describing the most recent state of the environment." col. 35, lines 19-26, "At the end of the profiling, COIGN writes the summary log of inter-component communication to a file for later analysis. In addition to information about the number and sizes of messages and components in the application, the profile log also contains information used to classify components and to determine pair-wise component location

constraints. Log files from multiple profiling executions can be combined and summarized during later analysis").

Per claims 5, and 11:

Hunt discloses:

- **the information about at least one prior compilation of the program includes a profile of the program implemented during the at least one prior compilation** (Hunt, col. 20, lines 17-27, "In an alternative embodiment, estimates of latency and bandwidth are periodically taken during execution of a distributed application. If the new estimates deviate beyond a preset threshold from previous estimates, the application is re-partitioned and distributed using the new estimates. In another embodiment, inter-unit communication is measured during distributed execution. If the communication characteristics of the distributed application deviate beyond a preset threshold from the communication characteristics used to determine the current distribution scheme, the distributed application is re-partitioned and re-distributed").

Per claims 6, and 12:

Hunt discloses:

- **the information about at least one prior optimization of the program includes a profile of at least one change made to the program during the at least one prior compilation** (Hunt, col. 39, lines 9-20, "Alternatively, COIGN can automatically decide when usage differs significantly from profiled scenarios, and silently enables

profiling for a period to re-optimize the distribution. The COIGN runtime already contains sufficient infrastructure to allow "fully automatic" distribution optimization. The lightweight version of the runtime, which relocates component instantiation requests to produce the chosen distribution, can count messages between components with only slight additional overhead. Run time message counts could be compared with relative message counts from the profiling scenarios to recognize changes in application usage").

Response to Arguments

8. Applicant's arguments filed 12/17/2007 have been fully considered but they are not persuasive.

Applicant argued:

Hunt fails to disclose or suggest optimizing program code. In fact, Hunt fails to relate to optimizing program code. Instead, Hunt relates to optimizing distribute computing by placing application at various locations on a computer network. None of the program codes of any Hunt's application are optimized. Instead, Hunt merely determines where to place the applications so that they are most effective.

Examiner responses:

Examiner disagree applicant arguments. Hunt does teach optimizing program code (Hunt, col. 30, lines 42-67 through col. 31, lines 1-25, "Although the foregoing techniques for dynamic classification have been described with reference to classification of application units in an

automatic distributed portioning system, these techniques are applicable outside of the field of automatic distributed portioning... For example, in the field of memory heap management, ... In the field of freeing unused memory...To balance the load among system resources in a computing system, a lifetime prediction for a program or data segment can be used to optimally allocate system resources...In general, the foregoing techniques for dynamic classification are applicable to profile-based optimization systems … Generally, there are three classes of solutions to accomplish this task according to the present invention: modify the application's source code, modify the application's binaries prior to execution, or manipulate the application's execution through run-time intervention. … Referring to code fragment 280 using call replacement in application source code, calls to the COM instantiation functions can be replaced with calls to the instrumentation by modifying application source…Using call replacement in application binary code (281), calls to the COM instantiation functions can be replaced with calls to the instrumentation by modifying application binaries…" emphases added). Hunt teaches dynamic classification and outside of the field of automatic distributed portioning are applicable to profile-based optimization systems that generally accomplish by three solutions: modify the application's source code, modify the application's binaries prior to execution, or manipulate the application's execution through run-time intervention. The modification in the source code and binaries (object code), and manipulate execution are interpreted to optimizing program code. Thus, Hunt teaches optimizing program code of the program.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136 (a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

1. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anna Deng whose telephone number is 571-272-5989. The examiner can normally be reached on Monday to Friday 9:30 AM - 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Anna Deng/

Examiner, Art Unit 2191

03/11/2008

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191